Rule CIC343: LOC=BELOW request initially failed and was retried

Finding: The CICS Shared Temporary Storage (TS) Queue Server statistics showed that a high percent of LOC=BELOW storage requests initially failed to obtain the requested storage, and requests were retried after merging adjacent small free areas to form larger areas.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of the CICS region. However, it could be a warning of a pending HIGH IMPACT on the performance of the CICS region.

Logic flow: This is a basic finding, based on an analysis of the data. The finding applies only with CICS/Transaction Server for OS/390 or for z/OS.

Discussion: Data items in shared temporary storage are kept in queues whose names are assigned dynamically by the program storing the data. These shared temporary storage queues are stored in *named pools* in an MVS coupling facility. A shared TS pool consists of an XES list structure on the coupling facility.

Access to a shared temporary storage pool by CICS transactions running in an AOR is through a TS *data sharing server* that supports a named pool. The data sharing server is started in its own region, by executing DFHXQMN. Various parameters are provided to DFHXQMN (POOLNAME, MAXQUEUES, BUFFERS, etc.) to allow tailoring of the data sharing server.

A data sharing server must be started on each MVS image for each pool defined in a coupling facility which can be accessed from that MVS image. All shared TS pool access is performed by cross-memory calls to the data sharing server for the named pool. The authorized cross-memory (AXM) page allocation services are used to manage server region storage after the server has been initialized.

During server initialization, the data sharing server acquires all of the available storage above the 16M line, as determined by the REGION size, then releases 5% of it for use by operating system services. This storage is referred to as *AXMPGANY* pool. The server also acquires 5% of the free storage below the line for use in routines which require 24-bit addressable storage. This storage is referred to as *AXMPGLOW* pool. Server statistics indicate how much storage is actually allocated and used within the storage areas above the 16M line (AXMPGANY pool) and below the 16M line (AXMPGLOW pool).

Storage is initially allocated from the pool using a bit map. For faster allocation, free areas are not normally returned to the pool but are added to a *vector of free chains* depending on the size of the free area (1 to 32 pages). When storage is being acquired, this vector is checked before going to the pool bit map.

If there are no free areas of the right size and there is not enough storage left in the pool, free areas in the vector are put back into the pool, starting from the smallest end, until a large enough area has been created. This action appears as a compress attempt in the statistics. If there is still insufficient storage to satisfy the request, the request fails.

If a task in the server region or a cross-memory request runs out of storage, this is likely to result in AXM terminating that task or request using a simulated ABEND with system completion code 80A to indicate a GETMAIN failure. Although the server can usually continue processing other requests, running out of storage in a critical routine can cause the server to terminate.

When requests in the AXMPGLOW pool are retried, this means that all free storage in the AXMPGLOW pool had been exhausted. This is not a problem, as such (the algorithm is designed to place storage buffers that are freed onto a vector of free chains). However, as the percent of requests that require a retry increases, overhead caused by the "compress attempt" to reduce fragmentation in the AXMPGLOW storage pool will correspondingly increase.

Shared temporary storage queue server storage statistics are available in MXG file CICXQ3. CPExpert uses data in CICXQ3 to calculate the percent of storage requests in the AXMPGLOW pool initially failed and were retried, using the following algorithm:

$$\textit{Percent AXMPGLOW storage requests retried} = \frac{S3LOWRQC}{S3LOWRQG}$$

where S3LOWRQC = Times a storage request initially failed and was retried
S3LOWRQG = Number of storage GET requests

CPExpert produces Rule CIC343 when the percent of storage requests in the AXMPGLOW pool initially failed and were retried is greater than the value specified by the **TSPCTLRC** guidance variable in USOURCE(CICGUIDE). The default value for the **TSPCTLRC** is .1, indicating that CPExpert should produce Rule CIC343 whenever more than one tenth percent of the storage requests in the AXMPGLOW pool initially failed and were retried.

Suggestion: If this finding is produced, you should consider the following alternatives:

- A possible cause of this situation is that the specific shared temporary storage queue is used for varying sizes of data items (if the shared TS queue contained data items with identical sizes, storage in the AXMPGLOW pool would not be fragmented). This situation could be caused by (1) a specific task using differing sizes for shared temporary storage items, or (2) different tasks using the shared TS queue and the different tasks have differing sizes of the shared temporary storage items.

  Consequently, one way to reduce the overhead caused by "compressing" the AXMPGLOW pool is to review the tasks using the shared TS queue. Determine whether multiple shared TS queues can be used (with each queue having a similar size).

- If this finding occurs often (or if a large percent of requests initially failed and were retried), this could be an indication that storage in the AXMPGLOW pool is becoming exhausted and requests might begin failing. Requests are retried only (1) if there are no free areas of the right size and (2) there is not enough storage left in the pool to satisfy the request. Either of these situations occurring frequently could indicate that storage in the pool is in danger of becoming exhausted. If storage is in danger of becoming exhausted, Rule CIC345 (LOC=BELOW had low percent minimum free storage) might be produced, but you might have altered the guidance for Rule CIC345 and Rule CIC345 might have been suppressed. If storage in the AXMPGLOW pool is becoming exhausted, you should consider the following alternatives:

  - Increase the amount of storage that is available for the shared TS server identified by this finding. Increasing the amount of storage can be accomplished by increasing the REGION parameter on the EXEC Job Control Language that starts the server.

    The queue server REGION parameter JCL should specify at least enough virtual storage for the specified number of buffers plus the storage used to process queue requests. Each buffer occupies a little more than 32K bytes, and each connected CICS region can have up to ten queue requests active at a time, each using 5K to 10K bytes. IBM suggests that the REGION size should allow at least 32K per buffer and 100K for each connected CICS region, plus a margin of about 10% for other storage areas.

  - Decrease the BUFFERS initialization parameter to leave more storage for request processing.

- You probably should take this action if Rule CIC332 (*Excess buffers defined for queue index buffer pool*) has been produced.

- You should normally NOT take this action if either Rule CIC330 (*High percent shared TS queue index buffers were in use*) or Rule CIC331(*High percent LRU activity in the TS queue index buffer pool*) had been produced, since these rules warn of potential pending storage shortage in the TS queue index buffer pool.

- Change the TSPCTLRC guidance variable in USOURCE(CICGUIDE) so Rule CIC343 is produced only when you wish to be aware of a different percent of storage requests in the AXMPGLOW pool that initially failed and were retried. Since Rule CIC343 provides an "early warning" of potential impending request failures, you normally should not change the TSPCTLRC guidance variable.

Reference: *CICS/TS for OS/390 Release 1.1*
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.2
CICS System Definition Guide: Section 3.4.3 (Defining TS server regions)

CICS/TS for OS/390 Release 1.3
CICS System Definition Guide: Section 4.2.2 (Defining TS server regions)

CICS/TS for z/OS Release 2.1
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

CICS/TS for z/OS Release 2.2
CICS System Definition Guide: Chapter 21 (Starting a temporary storage server)

